Data Science Initiatives

Reagan Moore
rwmoore@renci.org
University of North Carolina at Chapel Hill

- Data Intensive Cyber Environments Center
  - DataNet Federation Consortium
  - Research Data Alliance
- Renaissance Computing Institute
  - National Consortium for Data Science
  - iRODS Consortium
- School of Information and Library Science
  - LifeTime Library
  - Data science courses

Topics

1. DataNet Federation Consortium: federation across NSF cyber infrastructure projects
2. Progress towards a theory of data science: Basis for quantitative predictions about load and performance (Hao Xu)
3. Development of policies for managing protected data: pluggable rule engine in iRODS (Hao Xu)
4. iRODS Consortium: Industrial storage vendor provisioning of data management - packaging of iRODS with Seagate disk
5. Federation of repositories across federal agencies. Current applications of iRODS include NASA, NOAA, NOAO, and NSF awards.

DataNet Federation Consortium [1]

- University of North Carolina at Chapel Hill
  - Odum Institute - Dataverse (Social Science)
  - Institute for the Environment (Hydrology)
  - Renaissance Computing Institute - GENI (SDN Networks)
  - Data Intensive Cyber Environments Center (Data grids)
- University of California, San Diego
  - Science Observation Network – SciON (Sensor data)
  - Temporal Dynamics of Learning Center (Cognitive science)
- Arizona State University
  - Natural language processing (Computer science)
- University of Arizona
  - The iPlant Collaborative (Plant biology)
- University of Virginia
  - HydroShare (Hydrology)
- Drexel University
  - Semantic ontologies (Engineering)

Federation Mechanisms

- Claim three mechanisms are sufficient for federating existing data management systems
  - Tightly coupled federations
    - Shared name spaces
      - Federated data grids
  - Loosely coupled federations
    - Direct interaction using API of the remote system
      - Encapsulate published APIs in micro-services
  - Asynchronous federations
    - Indirect interaction
      - Communicate through a message bus

Federation of Data & Services

- Move data to remote service
  - Access remote service
  - Move data, apply service, and return results
- Move service to local data
  - Encapsulate service in virtual machine image (Docker)
  - Move service to local storage
  - Execute service as part of a local workflow

Theory of Data Science [2]

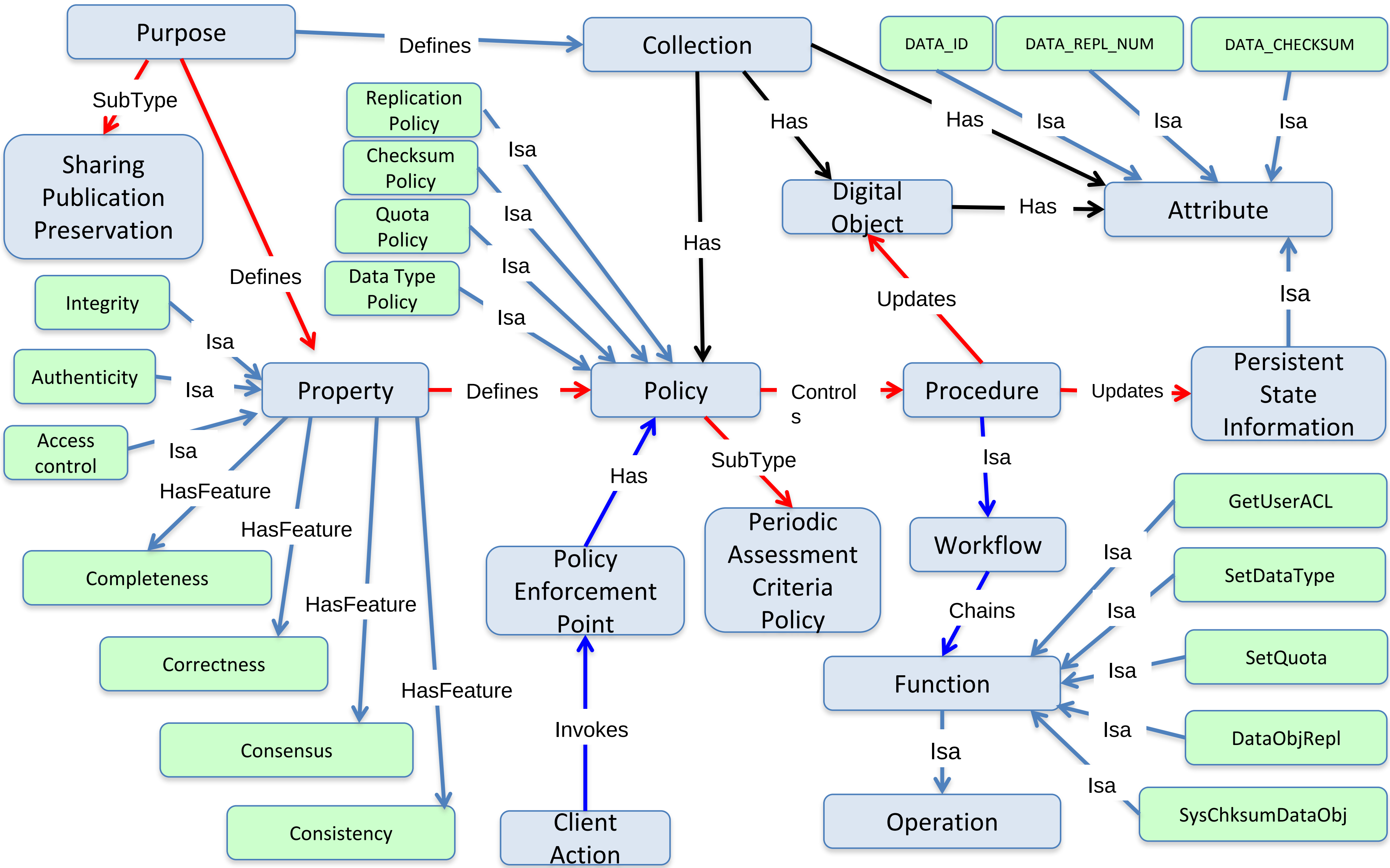
- A successful theory should support:
  - Characterization of a data management system through the changes to state information by operations
  - Identification of the assertions (conserved properties) maintained by the data management system
  - Prediction of the probability of success in maintaining the assertions in the presence of failure modes
  - Prediction of the sustainable workload
  - Identification of the assertions maintained across a federation
  - Prediction of the probability of success and sustainable workload of a federation

Required Infrastructure Components

- Policy-based system
  - Computer actionable rules
  - Computer executable procedures
- Persistent state information
  - Identification of the states changed by each operation
- Event tracking
  - For each operation, monitor the state changes

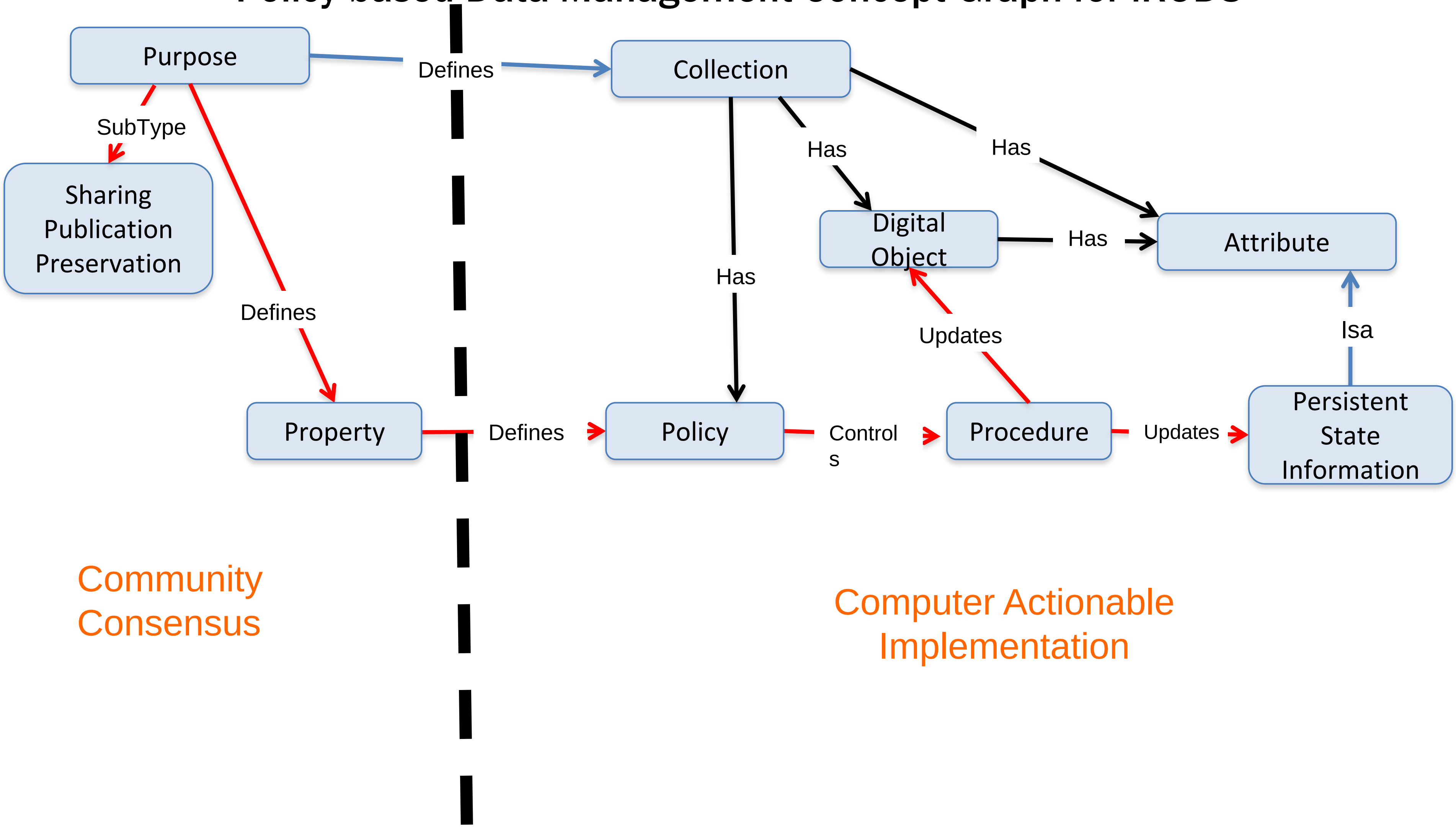
Policy Components - Conceptual Fundamentals

Policy-based Data Management Concept Graph for iRODS



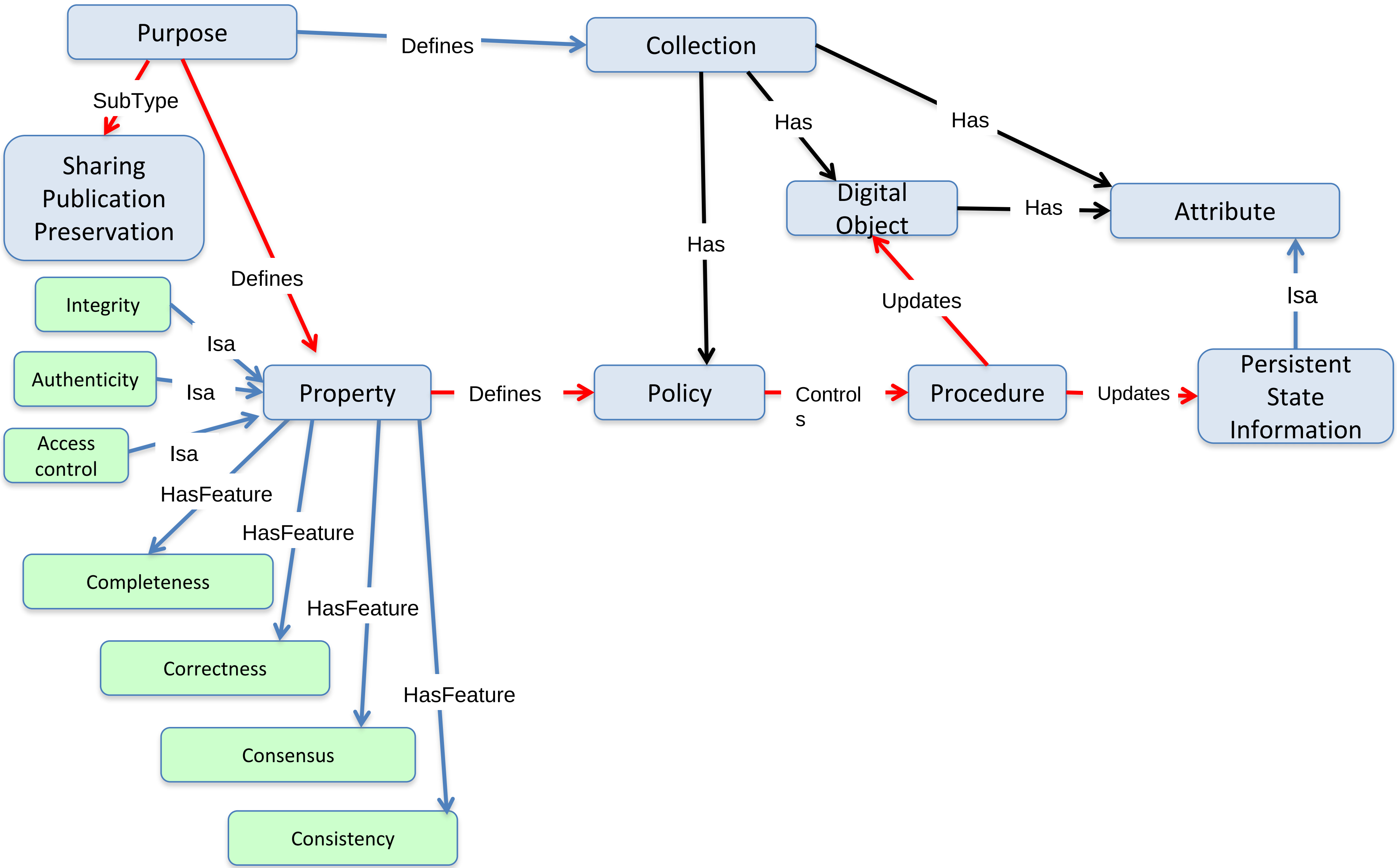
Policy Components - Conceptual Fundamentals

Policy-based Data Management Concept Graph for iRODS



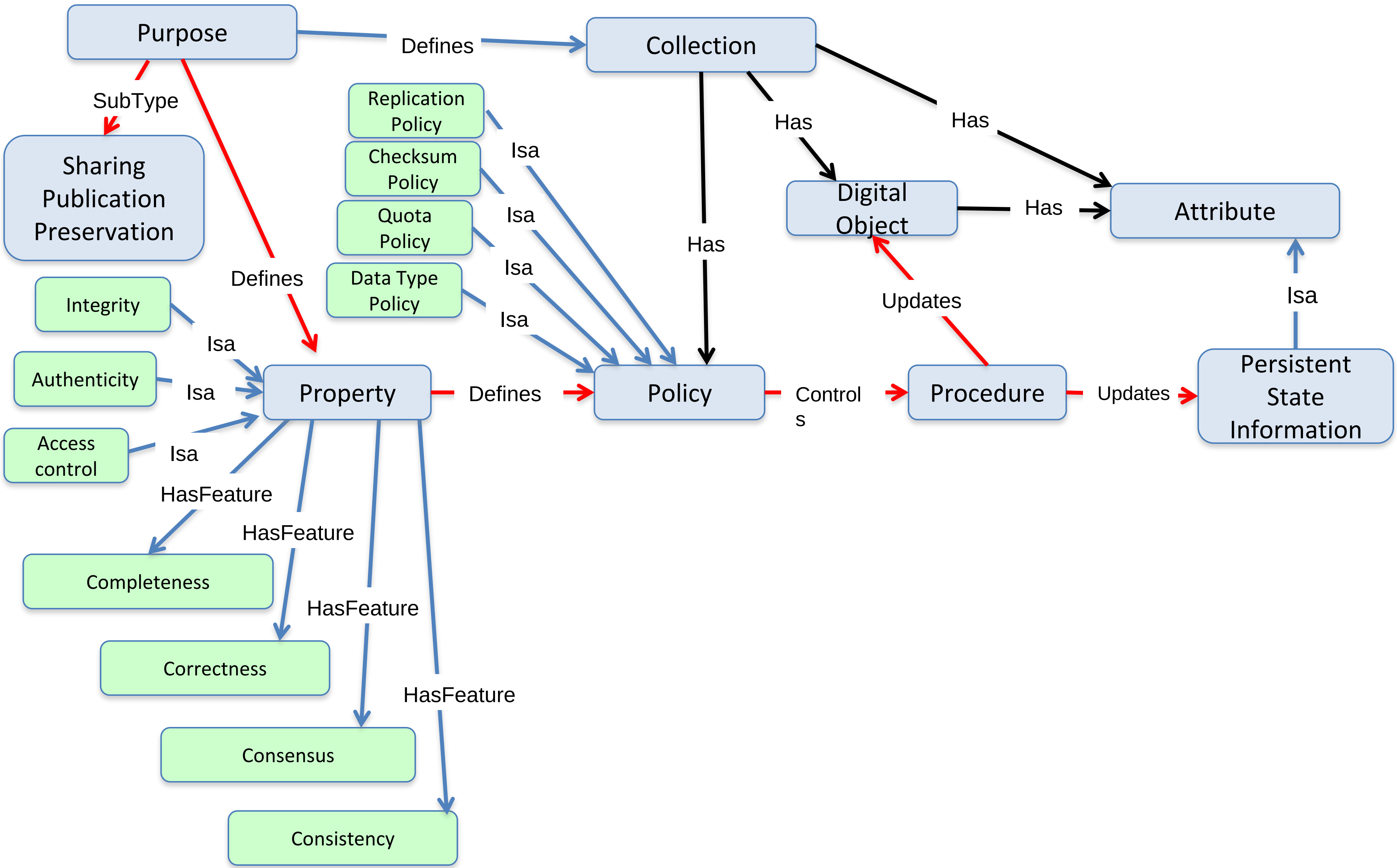
Policy Components - Conceptual Fundamentals

Policy-based Data Management Concept Graph for iRODS



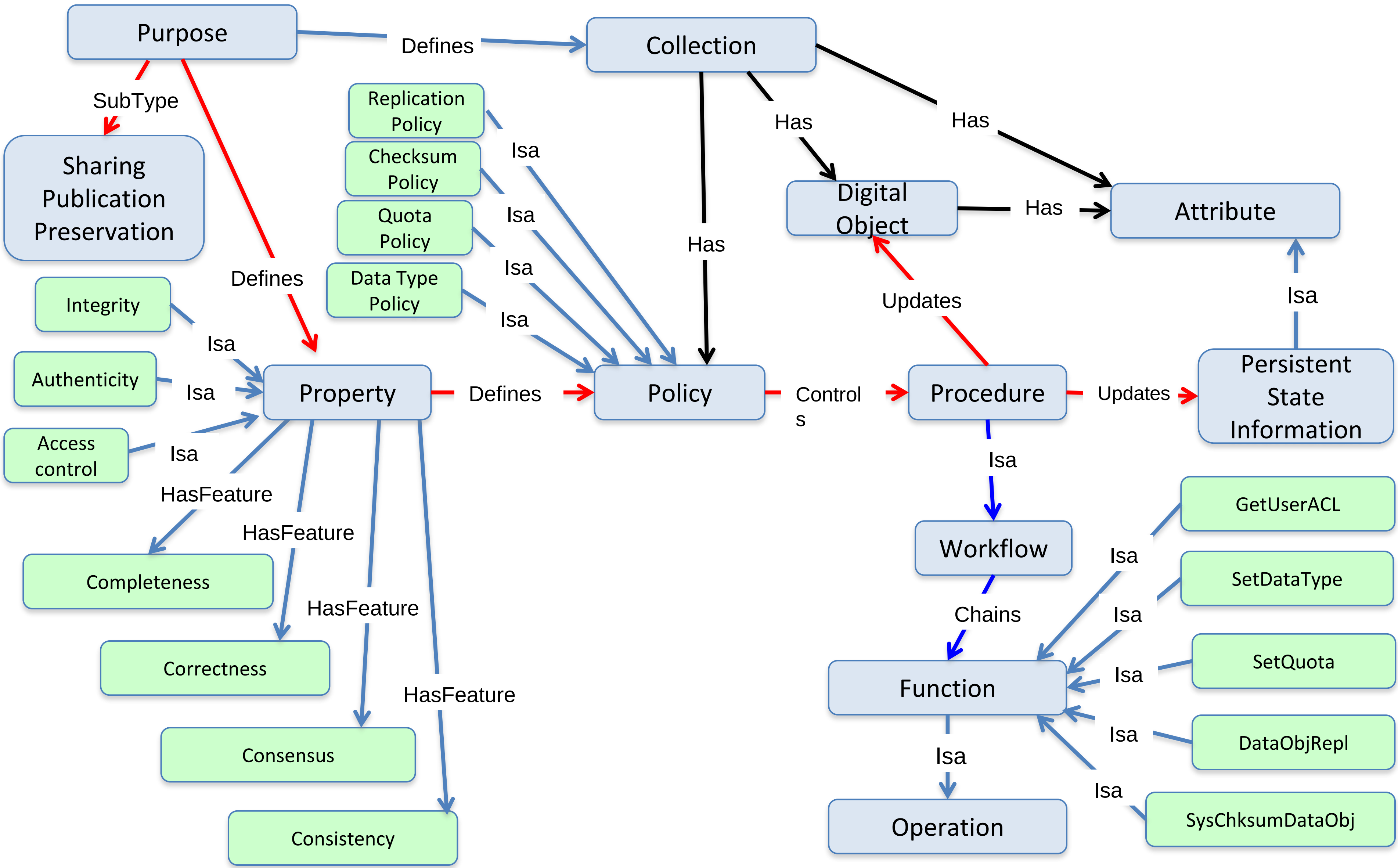
Policy Components - Conceptual Fundamentals

Policy-based Data Management Concept Graph for iRODS



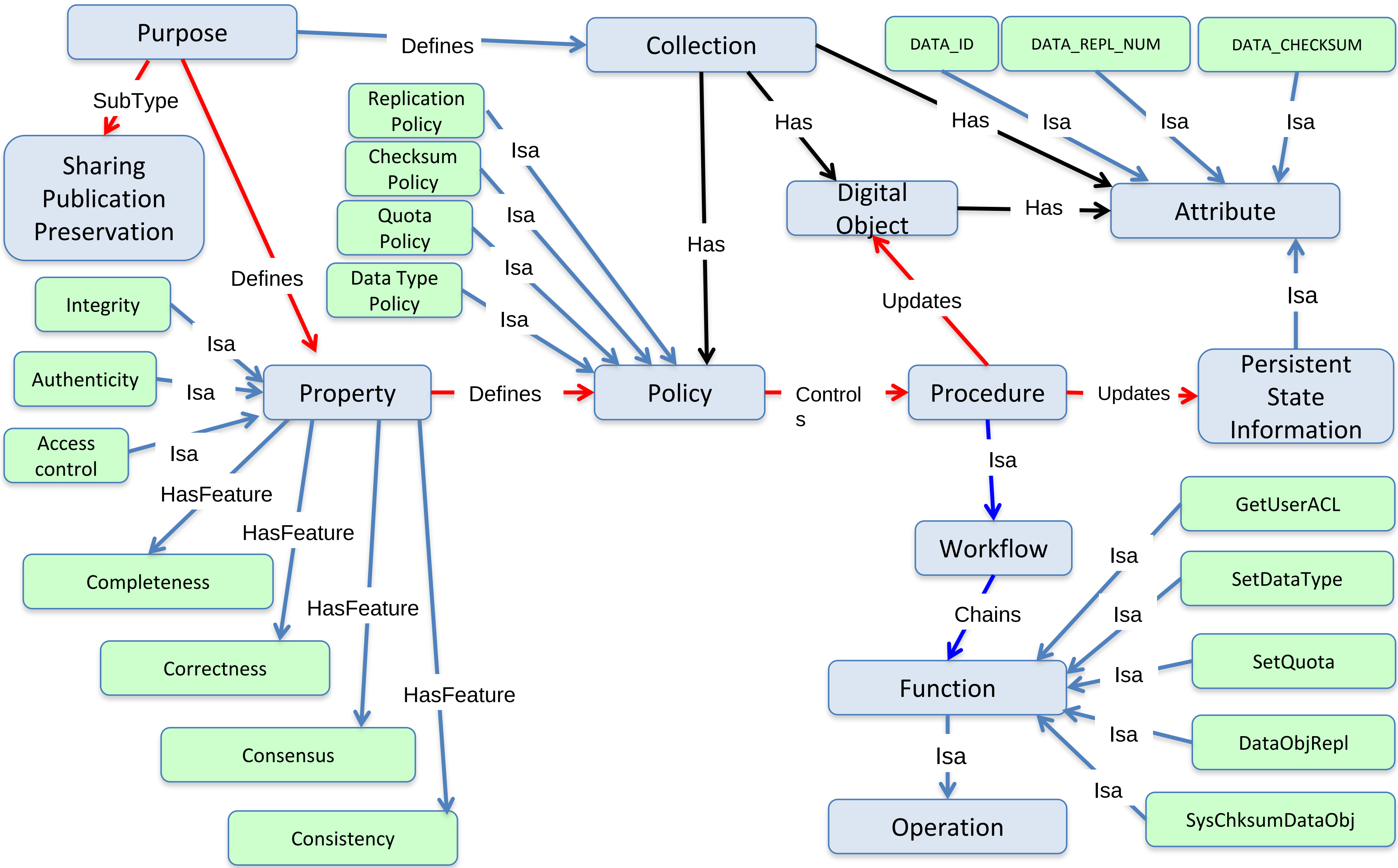
Policy Components - Conceptual Fundamentals

Policy-based Data Management Concept Graph for iRODS



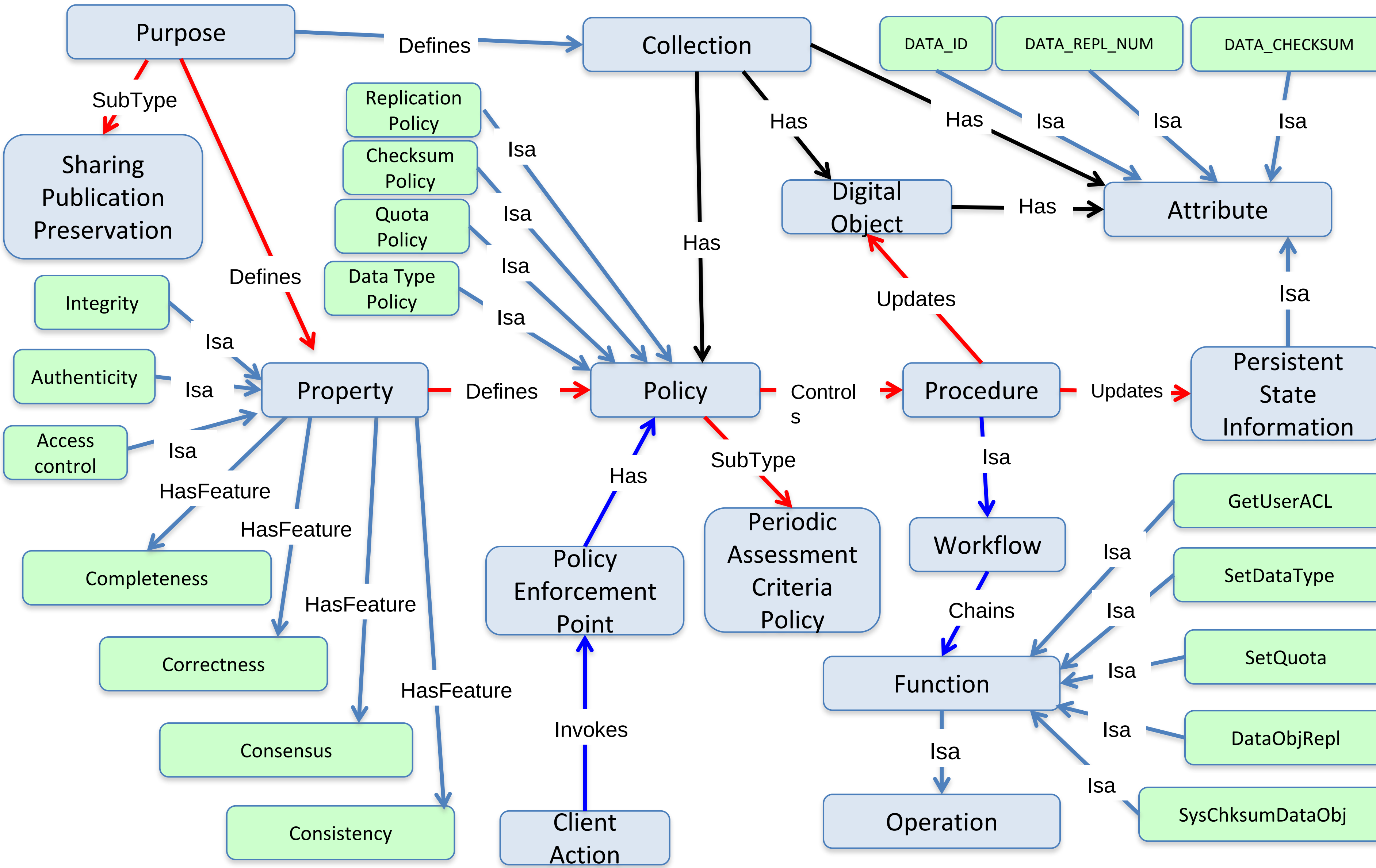
Policy Components - Conceptual Fundamentals

Policy-based Data Management Concept Graph for iRODS



Policy Components - Conceptual Fundamentals

Policy-based Data Management Concept Graph for iRODS



Data Science

- iRODS provides the required components
  - Operations defined in micro-services
  - State information updated on each operation
  - Policies control execution of the micro-services
    - Pre-process policy
    - Post-process policy
- Needed high performance tracking of events
  - Write rules in C++
  - Send event messages to external index

iRODS Pluggable Architecture

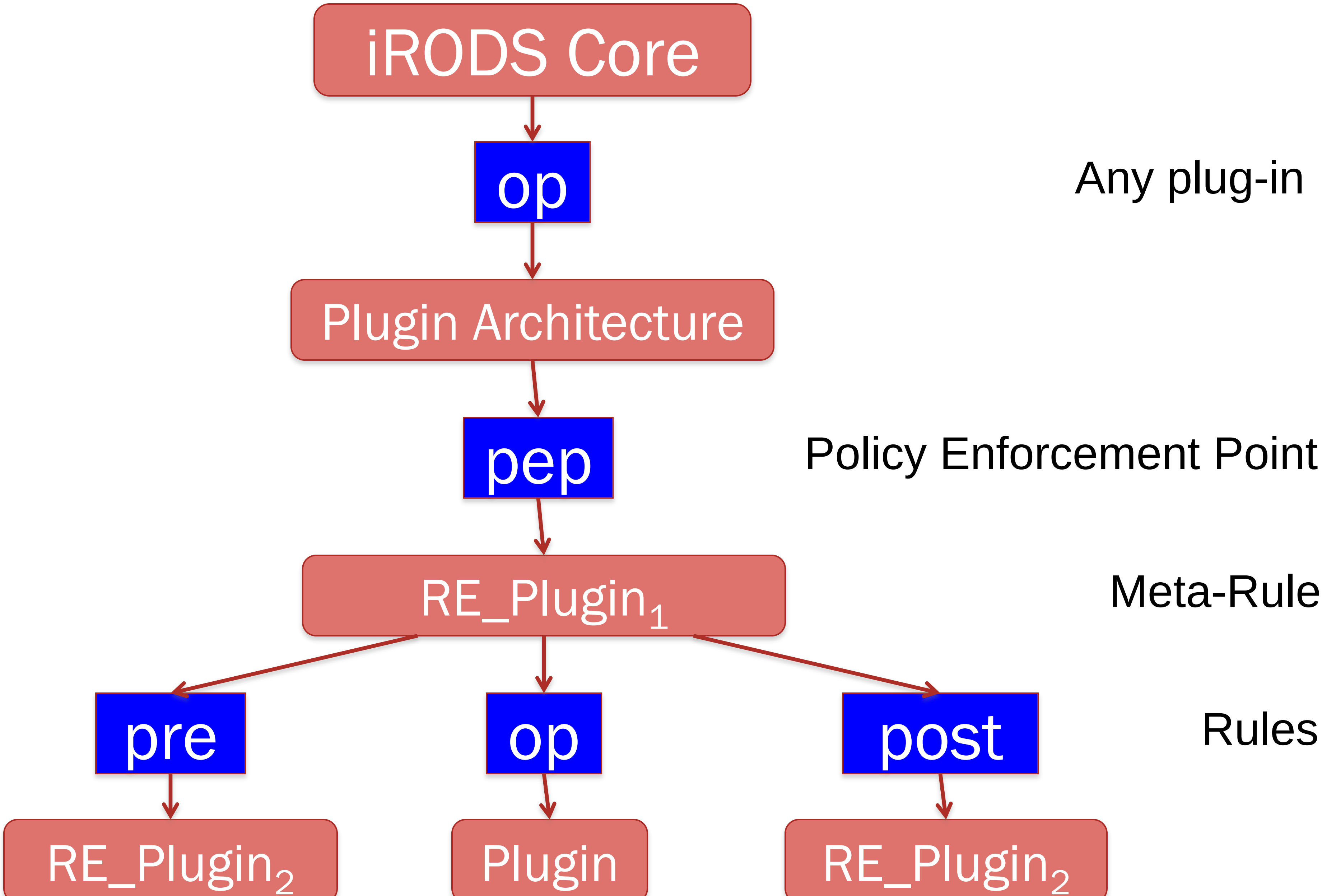
- Interactions with new technologies are encapsulated in plug-ins
  - API
  - Authentication systems (GSI, Kerberos)
  - Databases
  - Micro-services (curl)
  - Network
  - Storage systems (S3, WOS, HPSS)
  - Zonereport

Pluggable Architecture (Coposky)

- Plug-in new micro-services (iRODS 4.0)
  - Associate policy enforcement points with each plug-in
    - Pre-process policy
    - Post-process policy
- Pluggable Rule engine architecture (Hao Xu)
  - Support plug-ins for each rule language
    - iRODS, Python, JavaScript
  - Support meta-rule name spaces
    - Automatically add auditing rules to every micro-service invocation

Pluggable Rule Engine

- Supports 5 basic functions
  - Rule-engine-plugin-start
  - Rule-engine-plugin-stop
  - Rule-exists
  - Rule-execution
  - Callback
- Plugin typically is implemented in 100 lines of code
  - iRODS rule language, Python, Javascript, C++


iRODS Core
op
Any plug-in
Plugin Architecture
pep
Policy Enforcement Point
RE_Plugin1
Meta-Rule
pre
op
post
Rules
RE_Plugin2
Plugin
RE_Plugin2

Rule Name Space

- RE Plugin_1 provides extended namespace support for the translation to the default semantics.
- $\text{pep}[...] = \text{ns}_1\text{preop(args, env)} >> ...\ \text{ns}_n\text{preop(args, env)} >>$
  $\text{op(args,env)} >>$
  $\text{ns}_n\text{postop(args, env)} >> ...\ \text{ns}_1\text{postop(args, env)}$
- By default, we have namespace ns_1 = ””.
- We can add more namespaces. For example, for auditing ns_2 = ”audit ” or indexing ns_3 = ”index ” or security.
- For the audit namespace, pre and post file read PEPs:

  audit_pep_resource_read_pre

  audit_pep_resource_read_post

Policy Sets [3]

- Event auditing
- External indexing (events, metadata, text)
- Protected Data management
- Preservation
- Digital Library
- Data sharing

Protected Data

- Automated enforcement of 51 tasks, such as
  - Protected data type detection
  - Access approval flags
  - Encryption
  - Access control setting
  - Replication
  - Retention
  - Audit trail parsing for compliance
  - Verification of required access controls
  - Verification of integrity
  - Password constraints

iRODS User Group Meeting [4]

- iRODS Consortium – membership based sustainable infrastructure
  - Pharmaceutical companies
    - Genomics data grids
  - Storage vendors
    - Data management
    - Example – iRODS appliance integrates disk storage with pre-installed iRODS data grid (Seagate, DDN)
    - Can connect iRODS appliance to any data grid
    - Generalization of SAN technology

Federation [5]

- DataNet Federation Consortium pursuing federation across cyberinfrastructure projects and federal agencies
  - Data Infrastructure Building Blocks
    - GABBS – Geospatial Modeling and Analysis Building Blocks
    - Encapsulated service in Docker image
  - Data grids – NASA, NOAA, NSF, (EPA, NIH, NIEHS, NARA, LoC)
- Service federation
  - Through the Discovery Environment (iPlant) manage data movement and execution of encapsulated service on HPC resource (TACC)
  - Workflow structured objects – track provenance of workflows within iRODS

Virtualization of Data Flows

- Currently virtualize data collections and workflows
  - Manage their properties, access controls, provenance, naming, sharing, organization
- Can also virtualize data flows
  - Integration of Software Defined Networks and Policy-based data management
  - Enables re-execution of data flows, access controls on data flows, sharing, caching, assignment of I/O streams, event tracking, access by collection/file name

Development

- iRODS Consortium
  - iRODS release 4.1.3
  - Pluggable architecture – Jason Coposky
- DICE Center - DFC
  - Workflow structured objects – Arcot Rajasekar
  - Pluggable rule engine – Hao Xu
- RENCI
  - GENI – Shu Huang, Yufeng Xin
- Project support from:
  - NSF DataNet Federation Consortium Grant OCI 0940841

Federation Architecture